Automaty skończone

Wojciech Myszka

Streszczenie

pierwszy ustęp
drugi ustęp

Spis treści

Definicje **1**

Maszyna Turinga **2**

Przykład **2**

Budujemy maszynę Turinga **2**

Komplikacje zadania **3**

Wersja PDF dokumentu... **3**

Definicje

Tytułowy **automat skończony** to abstrakcyjny opis układu dynamicznego oparty na tablicy dyskretnych przejść między stanami. Konsekwencją tego opisu jest to, że czas biegnie w sposób dyskretny.

Układ statyczny to taki układ, którego zachowanie w chwili t zależy jedynie od wartości zmiennych wejściowych w chwili t (to znaczy nie zależą) od historii układu (wejść w chwilach poprzednich, zachowaniu układu w chwilach poprzednich).

Układ dynamiczny to taki układ, którego zachowanie w chwili t zależy od stanu układu i wymuszeń (wejść) w chwili t oraz w chwilach poprzednich.

Układów w pełni statycznych praktycznie nie ma. Ale bardzo często stosujemy ten opis w przypadku wolno-zmiennych wymuszeń. Układem dynamicznym jest, na przykład, jadący drogą samochód[1].

[1] Z tą konieczną uwagą, że jadącego samochodu nie opisuje się za pomocą automatu skończonego. Rzeczywiste

Maszyna Turinga

O maszynie Turinga mówiłem na na wykładzie z Technologii Informacyjnych. Konkluzje (zwane Tezą Churcha-Turinga) mówią, że każdy algorytm może być opisany za pomocą maszyny Turinga. Z drugiej strony wiemy (patrząc na formalny opis maszyny Turinga), że jest to szczególny przypadek automatu skończonego.

Przykład

Rozważmy działanie algorytmu, który ma dokonać konwersji ciągu znaków ASCII na liczbę i równocześnie zareagować na wszystkie możliwe sytuacje, które powodują, że ciąg znaków liczba nie jest.

Zastanówmy się jak może wyglądać **poprawna** (z punktu widzenia języka C) stałą dziesiętna.

1. Najpierw może być dowolna liczba „białych odstępów”[2].
2. Później znak (`-` lub `+`); gdy nie ma znaku zakłada się, że liczba jest dodatnia.
3. Później zaczynają się cyfry…

To, o czym trzeba pamiętać, to to, że **dowolna** kombinacja (białych) spacji, cyfr, znaków `+` i `-` nie tworzy poprawnej liczby. Na przykład `+123` liczbą jest, ale `-`*odstęp*`123` liczbą **nie jest**, bo po znaku dodano nadmiarowy odstęp. Nie jest też, na przykład, liczbą ciąg `12-3+`.

Patrząc okiem z daleka, od razu rozpoznajemy co liczbą jest, a co nią ie jest. Komputer pozbawiony jest takiej możliwości[3] i musi znaki analizować sekwencyjnie.

Budujemy maszynę Turinga

Najpierw alfabet znaków:

- cyfry od 0 do 9,
- znak o kodzie ASCII 0 (`NULL`),
- znak liczby (`+` lub `-`)[4],
- (białe) odstępy,
- wszystkie inne znaki.

Taśma to tablica tekstowa[5].

układy dynamiczne opisujemy zazwyczaj za pomocą równań różniczkowych.

[2]Biały odstęp (ang. *white space*) to każdy znak ASCII, który nie zostawia widocznego śladu na papierze (podczas wydruku). Należą do nich znaki odstępu (ASCII 32), tabulacji (ASCII 9), przejścia do nowej linii — generowany przez klawisz Enter (LF — *line feed* — czyli ASCII 10 lub CR — *carriage return* — czyli ASCII 13), przejście do nowej strony (FF — *form feed* — ASCII 12).

[3]Oczywiście można zacząć rozważać algorytmy Sztucznej Inteligencji systemu wyposażonego w kamerę; pytanie jest takie: *Jak będzie tam zaimplementowany algorytm rozpoznawania liczb?*

[4]*Znak* w sensie angielskojęzycznego *sign*.

[5]Sugeruję, nie korzystać z funkcji czytania danych z klawiatury.

Stan początkowy, to ten, w którym znajdujemy się przed przeczytaniem jakiegokolwiek znaku.

Diagram przejść. Najpierw opisowo

1. W stanie początkowym ignorujemy dowolną liczę (białych) odstępów;
   - gdy znajdziemy cyfrę — przechodzimy do stanu **Widziałem cyfrę**;
   - gdy znajdziemy znak — przechodzimy do stanu **Widziałem znak**;
   - każdy inny znak przechodzimy do stanu sygnalizacji **Błąd**.
2. W stanie **Widziałem znak** każdy inny kod ASCII niż odpowiadający cyfrze oznacza przejście do stanu sygnalizacji **Błąd**. Pamiętajmy, po znaku **musi** wystąpić cyfra.
3. W stanie **Widziałem cyfrę**:
   - napotkanie znaku ASCII o kodzie zero oznacza poprawne zakończenie programu[ˆ6];
   - napotkanie cyfry oznacza przejście do stanu **Widziałem cyfrę**;
   - napotkanie każdego innego znaku oznacza przejście do stanu sygnalizacji **Błąd**.

Nieudolnie narysowany przeze mnie diagram przejść znajduje się w instrukcji laboratoryjnej Laboratorium 10: „Maszyna stanow”.

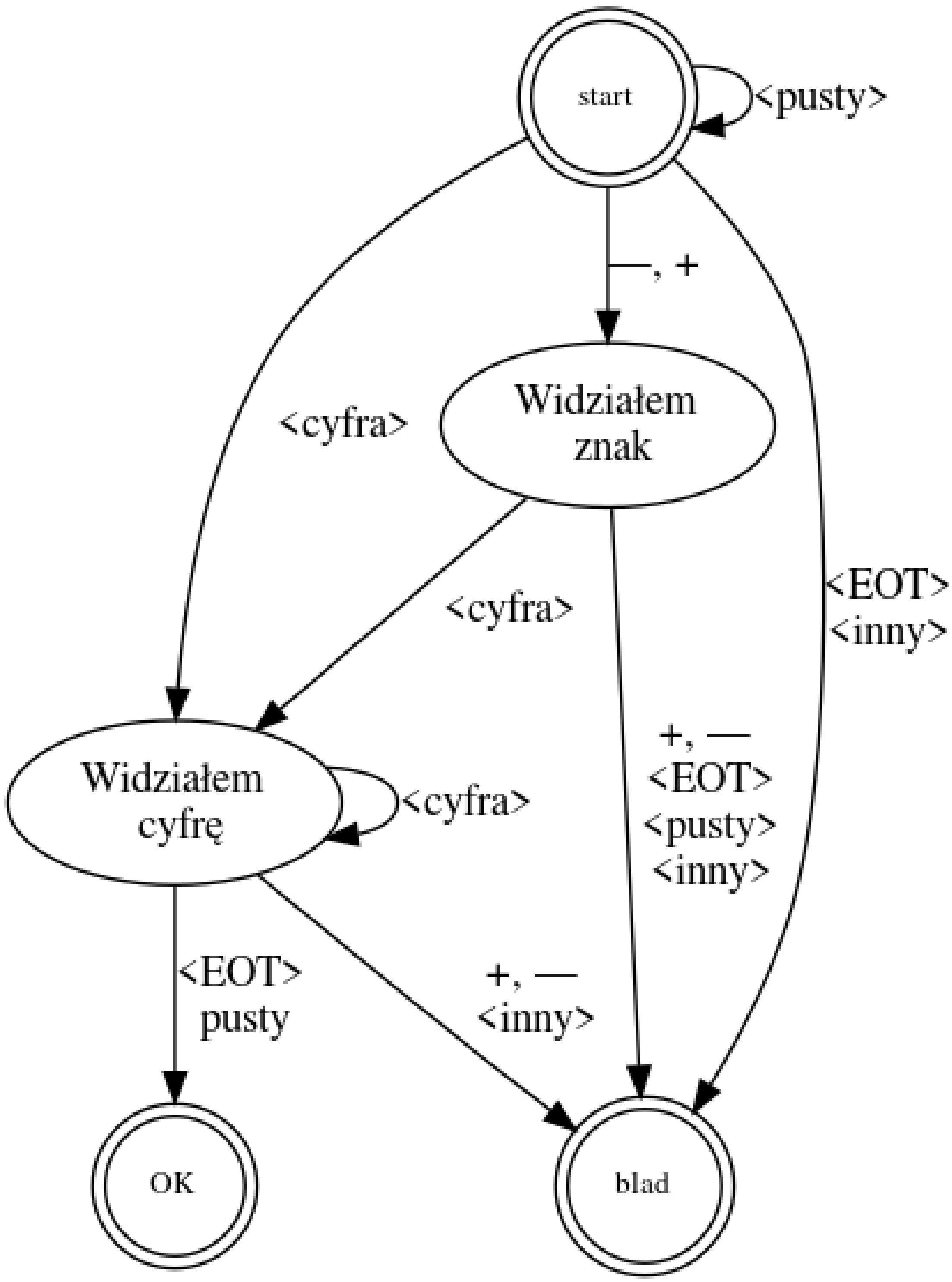
Wydaje mi się, że próba takiego spojrzenia na algorytm może okazać się przydatna.

Komplikacje zadania

1. Rozpoznawanie liczb całkowitych w formatach
   - dziesiętnym
   - ósemkowym (pierwsza cyfra 0)
   - szesnastkowym (pierwsze znaki to `0x` lub `0X`)
2. Rozpoznawanie liczb zmiennoprzecinkowych w podstawowych formatach:
   - `±x.xxxxx` (`x` cyfra dziesiętna)
   - `±x.xxxxE±xxx` (`E` może być wielkie lub małe)

Wersja PDF dokumentu…

…jest również dostępna.



Rysunek 1: Diagram przejść dla maszyny Turinga rozpoznającej liczby